WHAT IS CLAIMED IS:

1. A design apparatus compiled on a computer environment for generating from a behavioral description of a system comprising at least one digital system part, an implementable description for said system, said behavioral description being represented on said computer environment as a first set of objects with a first set of relations therebetween, said implementable description being represented on said computer environment as a second set of objects with a second set of relations therebetween, said first and second set of objects being part of a design environment, and wherein said first and second set of objects are part of a single design environment.

2. The design apparatus of Claim 1 wherein said first and second set of objects are part of a single design environment.

3. The design apparatus of Claim 1, wherein said design environment comprises an Object Oriented Programming Language.

4. The design apparatus of Claim 3, wherein said Object Oriented Programming Language is C++.

5. The design apparatus of Claim 1, wherein said design environment is an open environment wherein new objects can be created.

6. The design apparatus of Claim 1, wherein at least part of the input signals and output signals of said first set of objects are at least part of the input signals and output signals of said second set of objects.

7. The design apparatus of Claim 1, wherein at least part of the input signals and output signals of said behavioral description are at least part of the input signals and output signals of said implementable description.

8. The design apparatus of Claim 1, wherein said first set of objects has first semantics and said second set of objects has second semantics.

9. The design apparatus of Claim 8, wherein said first semantics is a data-vector model and/or a data-flow model.

10. The design apparatus of Claim 1, wherein the behavioral description includes a structure-free description.

11. A design apparatus compiled on a computer environment for generating from a behavioral description of a system comprising at least one digital system part, an implementable description for said system, said behavioral description being represented on said computer environment as a first set of objects with a first set of relations therebetween, said implementable description being represented on said computer environment as a second set of objects with a second set of relations therebetween, said first and second set of objects being part of a design environment, wherein said first and second set of objects are part of a single design environment, wherein said first set of objects has first semantics and said second set of objects has second semantics, and wherein said second semantics is a signal flow graph (SFG) data structure.

12. The design apparatus of Claim 11, wherein the impact in said implementable description of at least a part of the objects of said second set of objects is essentially the same as the impact in said behavioral description of at least a part of the objects of said first set of objects.

13. The design apparatus of Claim 11, further comprising means for simulating the behavior of said system, said means simulating the behavior of said behavioral description, said implementable description or any intermediate description therebetween.

14. The design apparatus of Claim 11, wherein at least part of said second set of objects is derived from objects belonging to said first set of objects.

15. The design apparatus of Claim 11, wherein said implementable description is at least partly obtained by refining said behavioral description.

16. The design apparatus of Claim 11, wherein said implementable description is an architecture description of said system.

17. The design apparatus of Claim 16, further comprising means for translating said architecture description into a synthesizable description of said system, said synthesizable description being directly implementable in hardware.

18. The design apparatus of Claim 17, wherein said hardware is a semiconductor chip.

19. The design apparatus of Claim 11, further comprising means to derive said first set of objects from a vector description describing said system as a set of operations on data vectors.

20. The design apparatus of Claim 19, wherein said vector description is a MATLAB description.

21. The design apparatus of Claim 11, further comprising means for simulating statically or demand-driven scheduled dataflow on said dataflow description.

22. The design apparatus of Claim 11, further comprising means for clock-cycle true simulating said digital system using said dataflow description and/or one or more of said SFG data structures using an expectation-based simulation.

23. The design apparatus of Claim 11, wherein the behavioral description includes a structure-free description.

24. A method of designing a system comprising at least one digital part, comprising refining, wherein a behavioral description of said system is transformed into an implementable description of said system, said behavioral description being represented as a first set of objects with a first set of relations therebetween and said implementable description being represented as a second set of objects with a second set of relations therebetween, and wherein said refining comprises translating behavioral characteristics at least partly into structural characteristics.

25. The method of Claim 24, wherein the behavioral description includes a structure-free description.

26. The method of Claim 24, further comprising simulating in which the behavior of said behavioral description, said implementable description and/or any intermediate description therebetween is simulated.

27. The method of Claim 24, wherein said refining comprises the addition of new objects, permitting interaction with existing objects, and adjustments to said existing objects allowing said interaction.

28. The method of Claim 24, wherein said refining is performed in an open environment and comprises expansion of existing objects.

29. The method of Claim 24, wherein said refining comprises first refining, said first refining comprising:

determining the input vector lengths of input, output and intermediate signals;

determining the amount of parallelism of operations that process input signals to output signals;

determining actors, edges and tokens of said data-flow model; and

determining the wordlength of said tokens.

30. The method of Claim 24, wherein said second set of objects with said second set of relations therebetween are at least partly derived from said first set of objects with said first set of relations therebetween.

31. The method of Claim 24, wherein objects belonging to said second set of objects are new objects, identical with and/or derived by inheritance from objects from said first set of objects, or a combination thereof.

32. The method of Claim 24, further comprising combining several SFG models with a finite state machine description resulting in an implementable description.

33. The method of Claim 32, further comprising transforming said implementable description to synthesizable code.

34. The method of Claim 33, wherein said synthesizable code is VHDL code.

35. A method of simulating a system, wherein a description of a system is transformed into compilable C++ code.

36. The method of Claim 35, wherein said description is an SFG data structure and said compilable C++ code is used to perform clock cycle true simulations.

37. The method of Claim 35, wherein the description includes a structure-free description.

38. A hardware circuit or a software simulation of a hardware circuit designed with the design apparatus of Claim 1.

39. A hardware circuit or a software simulation of a hardware circuit designed with the method of Claim 24.

40. A design apparatus compiled on a computer environment for generating from a behavioral description of a system comprising at least one digital system part, an implementable description for said system, said behavioral description being represented on said computer environment as a first set of objects with a first set of relations therebetween, said implementable description being represented on said computer environment as a second set of objects with a second set of relations therebetween, said first and second set of objects being part of a design environment, wherein said first set of objects has first semantics and said second set of objects has second semantics, and wherein said first semantics is a data-vector model and/or a data-flow model, wherein means for clock-cycle true simulating said digital system using said dataflow

description and/or one or more of said SFG data structures using an expectation-based simulation.

41. A method of designing a system comprising at least one digital part, comprising refining, wherein a behavioral description of said system is transformed into an implementable description of said system, said behavioral description being represented as a first set of objects with a first set of relations therebetween and said implementable description being represented as a second set of objects with a second set of relations therebetween, wherein said refining comprises first refining wherein said behavioral description is a data-vector model and is at least partly transformed into a data-flow model, and wherein said data-flow model is an untimed floating point data-flow model.

42. The method of Claim 41, wherein said refining further comprises second refining wherein said data-flow model is at least partly transformed into an SFG model.

43. The method of Claim 41, wherein said second set of objects with said second set of relations therebetween are at least partly derived from said first set of objects with said first set of relations therebetween.

44. The method of Claim 41, wherein objects belonging to said second set of objects are new objects, identical with and/or derived by inheritance from objects from said first set of objects, or a combination thereof.

45. A method of designing a system comprising at least one digital part, comprising refining wherein a behavioral description of said system is transformed into an implementable description of said system, said behavioral description being represented as a first set of objects with a first set of relations therebetween and said implementable description being represented as a second set of objects with a second set of relations therebetween, wherein said refining comprises:

determining the input vector lengths of input, output and intermediate signals;

determining the amount of parallelism of operations that process input signals to output signals;

determining actors, edges and tokens of said data-flow model; and

determining the wordlength of said tokens.

46. A method of designing a system comprising at least one digital part, comprising:

refining, wherein a behavioral description of said system is transformed into an implementable description of said system, said behavioral description being represented as a first set of objects with a first set of relations therebetween and said implementable description being represented as a second set of objects with a second set of relations therebetween,

wherein said refining comprises first refining, wherein said behavioral description is a data-vector model and is at least partly transformed into a data-flow model,

wherein said refining further comprises second refining, wherein said data-flow model is at least partly transformed into an SFG model, and

wherein said SFG model is a timed fixed point SFG model.

47. The method of Claim 46, wherein said second set of objects with said second set of relations therebetween are at least partly derived from said first set of objects with said first set of relations therebetween.

48. The method of Claim 46, wherein objects belonging to said second set of objects are new objects, identical with and/or derived by inheritance from objects from said first set of objects, or a combination thereof.

49. A method of designing a system comprising at least one digital part, comprising:

refining, wherein a behavioral description of said system is transformed into an implementable description of said system, said behavioral description being represented as a first set of objects with a first set of relations therebetween and said implementable description being represented as a second set of objects with a second set of relations therebetween,

wherein said refining comprises first refining wherein said behavioral description is a data-vector model and is at least partly transformed into a data-flow model,

wherein said refining further comprises second refining wherein said data-flow model is at least partly transformed into an SFG model, and

combining several of said SFG models with a finite state machine description resulting in an implementable description.

50. The method of Claim 49, further comprising the step of transforming said implementable description to synthesizable code.

51. The method of Claim 49, wherein said synthesizable code is VHDL code.

52. A method of simulating a system, wherein a description of a system is transformed into compilable C++ code, wherein said description comprises the combination of several SFG data structures with a finite state machine description resulting in an implementable description, said implementable description being said compilable C++ code suitable for simulating said system as software.

53. A method of simulating a system, wherein a description of a system is transformed into compilable C++ code, wherein said simulating comprises a clock-cycle true simulation of said system being an expectation-based simulation using one or more SFG data structures, said expectation-based simulation comprising:

annotating a token age to every token;

annotating a queue age to every queue;

increasing token age according to the token aging rules and with the travel delay for every queue that has transported the token;

increasing queue age with the iteration time of the actor steering the queue; and

checking whether token age is never smaller than queue age throughout the simulation.